Title: Ant Colony Optimization based classifier in Performance Autotuning

Orio is a framework for automatically tuning the performance of codes written in different languages. It generates many tuned versions of the same operation using different optimization parameters, and performs an empirical search for selecting the best among multiple optimized code variants. There are different search algorithms that are used to select this optimal result like exhaustive, simplex, random search etc. In the process of finding the optimal output, these algorithms tests different combinations of the tuning parameters and runs several trials for getting the runtimes. So we thought of using this huge data to design a classification problem. We are using ant colony optimization algorithm (ACO) for this. First we went through the literature and found out an interesting approach called the Antminer. It classifies the given dataset using basic ant colony algorithm. We decided to use the data produced by current search algorithms as input to the ACO-based. We consider three classes of code versions based on their performance (best-performing, worst-performing, and the rest). The performance results are given to the process, where classes are predefined based on attributes and a classifier (here Antminer) allocates data samples to classes. For the attributes, we consider various code specific values (number of loops and number of nested loops in a code), which can be used to identify other codes with similar characteristics. After the classification is performed on our training sets, we evaluate its effectiveness by using it to guide the autotuning of codes that were not used as part of the training set.